Sistemi di numerazione e rappresentazione binaria

CORSO DI ARCHITETTURA DEGLI ELABORATORI E LABORATORIO– MODULO LABORATORIO

GABRIELLA VERGA

Numeri virgola fissa

- Un bit di segno
- Una porzione di bit fissa per la parte intera
- Una porzione di bit fissa per la parte decimale

Con 32 bit, per un intero con segno in complemento a due, possono essere codificati i numeri nell'intervallo da a $\mathbf{-2^{-31}, +2^{31}}$-1, ovvero in decimale da a $\mathbf{-10^{-10}, +10^{10}}$ circa. Per numeri frazionari si può considerare che la virgola sia presente (fissa) tra il bit_{31} e il bit_{30} (appena dopo il bit di segno), si può quindi rappresentare un valore piccolo come $\mathbf{10^{-10}}$.

Per numeri piccoli va bene, per numeri grandi no ➔ ***Intervallo non sufficiente per calcoli scientifici***

Numeri virgola mobile

Per rappresentare numeri sia grandi che piccoli si usa la rappresentazione in **virgola mobile**.

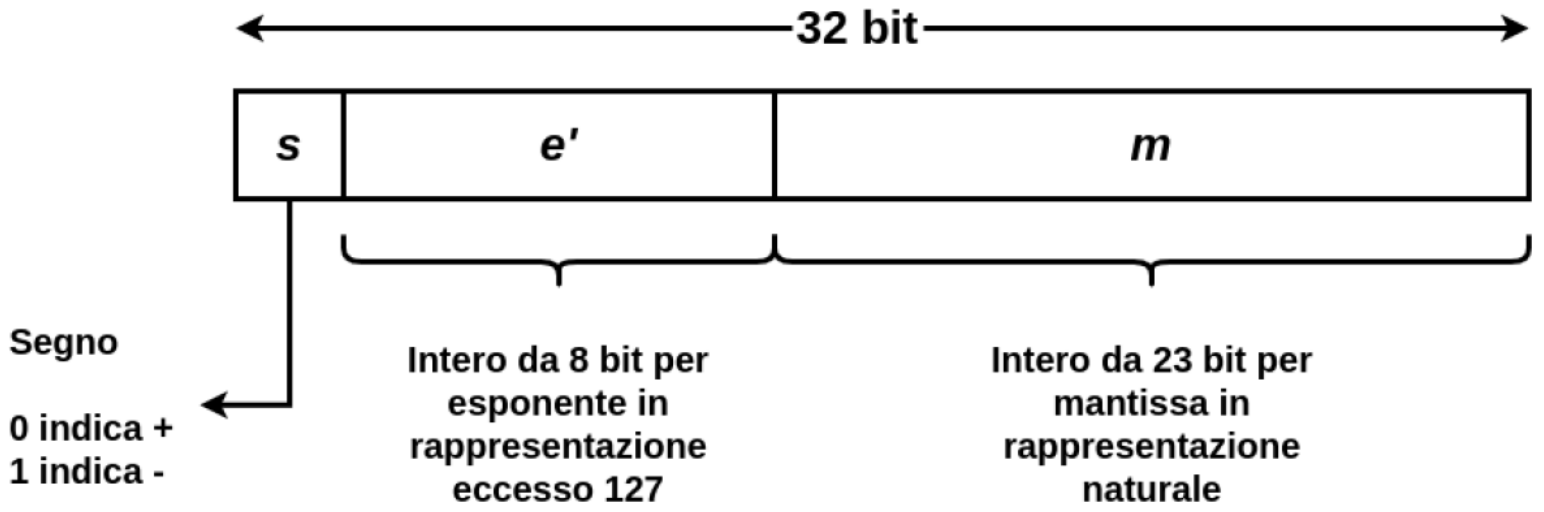
Un numero binario in virgola mobile può quindi essere rappresentato:

- un **SEGNO** *s* per il numero
- la **MANTISSA** *m* (bit significativi *escluso il bit più significativo*)
- un **ESPONENTE** *e* con segno in base 2

Standard

- Standard IEEE 754 numeri 32 bit:
  - ✓ 1 bit (segno)
  - ✓ 23 bit (mantissa)
  - ✓ 8 bit (esponente)
- Standard IEEE 754 numeri 64 bit
  - ✓ 1 bit (segno)
  - ✓ 52 bit (mantissa)
  - ✓ 11 bit (esponente)

Standard IEEE 754 numeri 32 bit



e' = e + 127

Intervallo esponente: **$-126 \leq e \leq 127$**

Fattore di scala nell'intervallo: **$[2^{-126}, 2^{127}]$ in decimale $[\pm 10^{-38}, \pm 10^{38}]$**

Esempio 1

-118,5

1) 118 = 1110110; 0,5 = 10 ➔ 1110110,10
2) e = 6 ➔ e + 127 = 133 ➔ 10000101

In definitiva:

- s = 1
- e’ = 10000101
- m = **11011010**00000000000000

1 10000101 1101101000000000000000

Esempio 2

-109,78125

1) 109 = 1101101; 0,78125 = 11001 ➔ 1101101,11001
2) e = 6 ➔ e + 127 = 133 ➔ 10000101

In definitiva:

- s = 1
- e’ = 10000101
- m = **10110111001000000000000**

1 10000101 10110111001000000000000

Esempio 3

23,6875

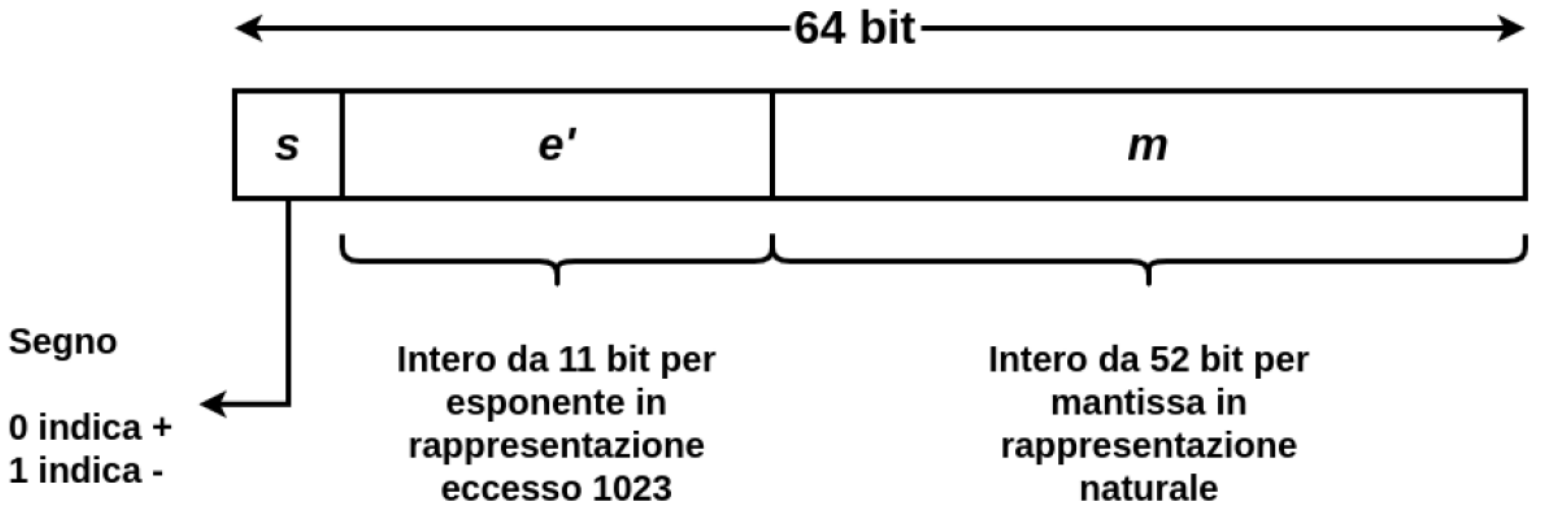
1) 23 = 10111; 0,6875 = 1011 ➔ 10111,1011
2) e = 4 ➔ e + 127 = 131 ➔1000 0011

In definitiva:

- s = 0
- e’ = 1000 0011
- m = **01111011000000000000000**

0 10000011 011110100000000000000

Standard IEEE 754 numeri 64 bit



e' = e + 1023

Intervallo esponente: **$-1022 \leq e \leq 1023$**

Fattore di scala nell'intervallo: **$[2^{-1022}, 2^{1023}]$**

Esempio 4

Rappresentare in decimale il seguente numero binario:

a) 0 10000010 011010...0

Esempio 4

Rappresentare in decimale i seguenti numeri binari in formato a precisione singola:

a) 0 10000010 011010...0

Si ha:

- Segno 0 → positivo
- e': 130 e: 130 – 127 = 3
- m: 01101

Decimale: $1{,}01101 * 2^3 = 1011{,}01 = 11{,}25$

CODICE ASCII

- **Codice ASCII** (American Standard Code for Information Interchange)
- Rappresenta lettere, cifre decimali, punteggiatura e caratteri speciali
- Definito su **7 bit** → alfabeto di **2^7 = 128 elementi**
- Lettere e numeri con codici in ordine crescente

	Bit 654							
Bit 3210	000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	”	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	/	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Standard di internalizzazione

Necessita di codici più ricchi per gestire le diverse lingue con caratteri speciali, accenti, etc.

Standard internazionali:

- Famiglia **ISO 8859-x**: estendono il codice ASCII usando 8 bit (doppio dei simboli)
- **ISO/IEC 10646 (UCS)**: rappresentazione universale di caratteri che estende su più byte la ISO 8859
- Standard di codifica basati su UCS: come ad esempio **UNICODE** e **UTF-8**

Esercizi

1. Data la coppia di numeri 12 e 14
   I. Convertirli in numeri di 5 bit in complemento a 2
   II. Eseguire la somma
   III. Valutare se è avvenuto trabocco

2. Rappresentare il numero 25,45 come numeri binari a virgola mobile e formato a precisione singola (32 bit). Nella conversione approssimare il numero alla 4 cifra binaria dopo la virgola.

Esercizio 1

12 : 2	6	**0**
6 : 2	3	**0**
3 : 2	1	**1**
1 : 2	0	**1**

14 : 2	7	**0**
7 : 2	3	**1**
3 : 2	1	**1**
1 : 2	0	**1**

01110 ➔ 10001 + 00001= 10010

(-14)

12 = 01100 -14 = 10010	01100 + (12) 10010 = (-14) 11110 => -2	01100 + (12) 01110 = (14) 11010 (-6) => TRABOCCO

Esercizio 2

Valore assoluto in binario: 11001,0111

m: 10010111
e: 4 e': 127 + 4 = 131 = 10000011
segno: 0

0 10000011 100101110…0